

6. Dilbert

Program Name: Dilbert.java

Input File: dilbert.dat

Dilbert likes to play around with numbers and has developed an interesting process for manipulating positive integers. He wants to reverse integers and 12345 reversed in decimal form is 54321 which is just too simple. However, the binary form might be more interesting. The binary form of 12345 is **0011 0000 0011 1001** (separated into 4-bit groups for readability) and the reverse of the binary form would be **1001 1100 0000 1100** which has a decimal value of 38848, not 54321! But to make it even more interesting, lets discard leading 0s of the initial binary form before it is reversed. So, the modified binary form now becomes **11 0000 0011 1001** and when reversed produces **10 0111 0000 0011** which has a decimal value of 9987. Hmmm...

Can help Dilbert by writing program to verify his results?

Input: An unknown number of whitespace separated positive integers but no more than 100. Each integer is a single test case and will consist of a single positive integer that will not exceed the value 9×10^{18} .

Output: Each test case will produce 1 line of output containing the decimal equivalent of the reversed binary form followed by a single space and the reversed full binary form without leading 0s

Sample input:

```
10 12345 54321 65536 2903185746
17 1 257 2147483647 1073741984 41943185
```

Sample output:

```
Test Case: 1: 5 101
Test Case: 2: 9987 100111000000011
Test Case: 3: 35883 1000110000101011
Test Case: 4: 1 1
Test Case: 5: 1251528885 1001010100110001101000010110101
Test Case: 6: 17 10001
Test Case: 7: 1 1
Test Case: 8: 257 100000001
Test Case: 9: 2147483647 11111111111111111111111111111111
Test Case: 10: 41943041 10100000000000000000000000000001
Test Case: 11: 35913733 100010010000000000000000000101
```